

ARMY RESEARCH LABORATORY



Description of an Automated Tool for Evaluating a Tactical Communications Protocol

by María C. López

ARL-MR-509

March 2001

Approved for public release; distribution is unlimited.

20010403 081

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-MR-509**March 2001**

Description of an Automated Tool for Evaluating a Tactical Communications Protocol

María C. López

Computational and Information Sciences Directorate, ARL

Approved for public release; distribution is unlimited.

Abstract

Adaptive networking (AN) focuses on creating a logical and extendible framework for a multiyear effort in AN research. The U.S. Army Research Laboratory (ARL) designed an experiment to identify which communications protocol's parameters were more likely to have a relevant effect on network congestion. The software, called experiment control software (exp_driver), was developed to automatically execute the experimental design to test the protocol's parameters. Prior to the exp_driver existence, the experimental design for similar tests was executed with frequent operator intervention, requiring extra time for set up, and introducing human errors. This report provides a description of the exp_driver configuration for others who want to implement this software.

Acknowledgments

The author appreciates Ann Brodeen for helping with understanding the statistical terms and procedures mentioned in this report. Likewise, the author appreciates Fred Brundick and George Hartwig, Jr., for taking the time to answer questions related to the software language and the in-house tactical communications protocol, respectively.

INTENTIONALLY LEFT BLANK.

Table of Contents

	<u>Page</u>
Acknowledgments.....	iii
List of Figures	vii
1. Introduction.....	1
2. Software Configuration.....	2
3. Flow of Events	4
4. Hardware Configuration	6
5. Conclusions.....	7
6. References.....	9
Distribution List.....	11
Report Documentation Page	13

INTENTIONALLY LEFT BLANK.

List of Figures

<u>Figure</u>	<u>Page</u>
1. Template File With Shell Procedure Interaction.....	3
2. Flow of Events.	5
3. Example of a Hardware Configuration.	7

INTENTIONALY LEFT BLANK.

1. Introduction

Adaptive networking (AN) focuses on creating a logical and extendible framework for a multiyear effort in AN research. Advanced concepts to distribute information on the battlefield are considered. The automatic adjustment of the communications (commo) protocol's parameters or factors, such as message length and arrival rate, based on heuristics and continuous monitoring of network statistics, is one approach being pursued to improve real network throughput and minimize delay. The U.S. Army Research Laboratory (ARL) designed an experiment to identify which factors were more likely to have a relevant effect on network congestion [1]. The experimental setup consisted of various computers, referred to as remote nodes, connected to tactical net radios via modems, each running a scenario driver, and the commo protocol to be tested. Each remote node collected information on messages sent and received to analyze at a later time. The software called **exp_driver** was developed to automatically execute the experimental design to test the protocol's factors where k factors were combined and tested at two levels (low and high), implementing a 2^k factorial design [2]. The combination of factors and levels, or factor level combinations, form the design matrix. The design matrix was executed and replicated several times. The messages for the scenario driver for each factor level combination were distributed randomly in time according to a Poisson distribution.

Prior to **exp_driver** existence, the experimental design for similar tests was executed with frequent operator intervention. This required extra time for setup and introduced the possibility of human errors during the initialization phase and data reduction phase of a factor level combination.

The objective of this report is to provide a description of the experiment control software configuration for others who want to implement this software. It is written in the C programming language and uses X Window/Motif functions under the UNIX operating system.

2. Software Configuration

exp_driver is a menu-driven user interface written in C programming language [3] that uses the X Window library [4] and the Motif toolkit [5]. It coordinates all tasks necessary to execute the experimental design from a computer, herein referred to as the control node. Software execution, inputs, and factor level combinations for the test are controlled by **exp_driver**.

Among its tasks, **exp_driver** generates messages for the scenario driver, updates the factor level combinations, distributes the information to the remote nodes, and synchronizes the nodes' clocks. In addition, it starts and ends each factor level combination, retrieves all log files from the remote nodes to store on the control node, and computes measures of performance. To minimize blocking and possible input errors, **exp_driver** runs all factor level combinations without human intervention. The software is capable of executing independent replications of the design matrix automatically, with each replication using different random numbers, starting in the same initial state, and all statistical counters reset to zero.

exp_driver reads text files to obtain initial values that may vary depending on the experimental design. These text files contain values that need initialization prior to the factor level combination, such as factors and levels of interest, the number of replicates for each factor level combination, the number of replicates for the center point, the random number seeds to generate the desired message sets or scenarios, the number of tries for each message not delivered during the retry timeout, the node identification string, and the length of each factor level combination. Other values that are initialized are the name of the directories into which the software will store the data, the directories where shell procedure files that need execution are located, and values that are used by the data reduction software. The text files used for initialization may be modified either by editing the files prior to running **exp_driver**, or by menu selection before executing the experimental design.

The commo and scenario driver software on the remote nodes have their own input files and need to be updated prior to each factor level combination. The control node has a copy of these files (template files), which **exp_driver** updates and copies on the remote nodes. Template files are used whenever part of a file needs to be modified. Template files are similar to a form where the user fills in the blanks. The blanks are filled with the information for that specific factor level combination, and the file is copied either on the control or remote node where it will be

used during the test. Examples of this kind of file are the commo protocol input file loaded by the commo software to initialize the nodes' id, the window size and retry timeout (Figure 1a), and the nodename## file from which the scenario driver gets the message information to load messages into commo software.

exp_driver executes some of its tasks by invoking UNIX shell procedures [6]. On the remote node, it synchronizes clocks and starts and ends the execution of a factor level combination (Figure 1b). On the control node, it copies and retrieves files to and from the remote nodes (Figure 1a).

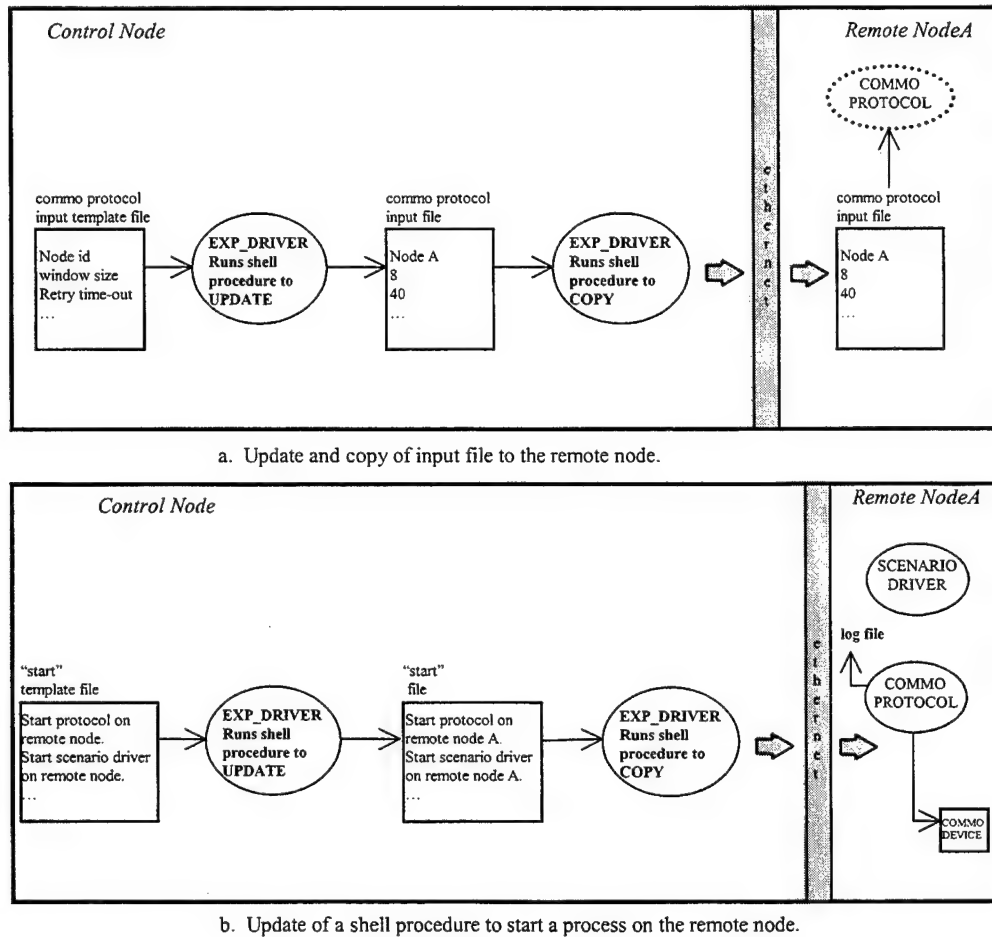


Figure 1. Template File With Shell Procedure Interaction.

While executing a factor level combination, each node collects data on a log file local to that node. The log files contain information on the messages and acknowledgments (ACKs) sent and received, as well as information on queues. The **data reduction** software is a set of C language programs that reformat log files and compute measures of performance. **Exp_driver** executes

UNIX shell procedures to invoke the data reduction software. For example, for each log file created by a factor level combination, **exp_driver** executes a shell procedure *process.s*, which invokes the C program *dr* with a set of arguments to create a file containing the messages transmitted during a particular factor level combination grouped into 1-min time intervals. The shell procedure *process.ack* invokes *dr* with different arguments and outputs all the ACKs generated during that factor level combination. The shell procedures that contain node information are updated using template files. The output of the data reduction software is formatted in a fashion suitable for a statistical analysis.

3. Flow of Events

When the experimenter selects automatic mode, **exp_driver** executes the first run or the next replication of the test. Following the steps on Figure 2, the data files for the scenario driver are generated first. Then, the commo protocol input files are updated with the current factor level combination information. The data and commo protocol input files are then copied onto the remote nodes. Clocks are synchronized by setting the time of the day on the remote nodes to the same time on the control node. Once the initial setup is completed, the test is ready to start. The software now executes a shell procedure to execute the commo and scenario driver software on the remote nodes. Once the scenario driver puts all messages for that factor level combination in the queue for transmission, **exp_driver** waits a specified amount of time to make sure the queue empties before the software is stopped. Once this time is up, all software on the remote nodes is stopped by running another shell procedure, the log files are copied onto the respective directory on the control node, and the data reduction software is applied to the log files on the control node. The **exp_driver** now continues with the next factor level combination, applying the procedure just described until all factor level combinations are completed. It continues with the center point factor level combinations in the same fashion until they are completed, ending the test.

The test may be stopped and/or continued through menu selection if any problems arise during a specific run. When the test is stopped in the middle of a run, that combination may be restarted or the next combination started, but it cannot be started in the middle of the run.

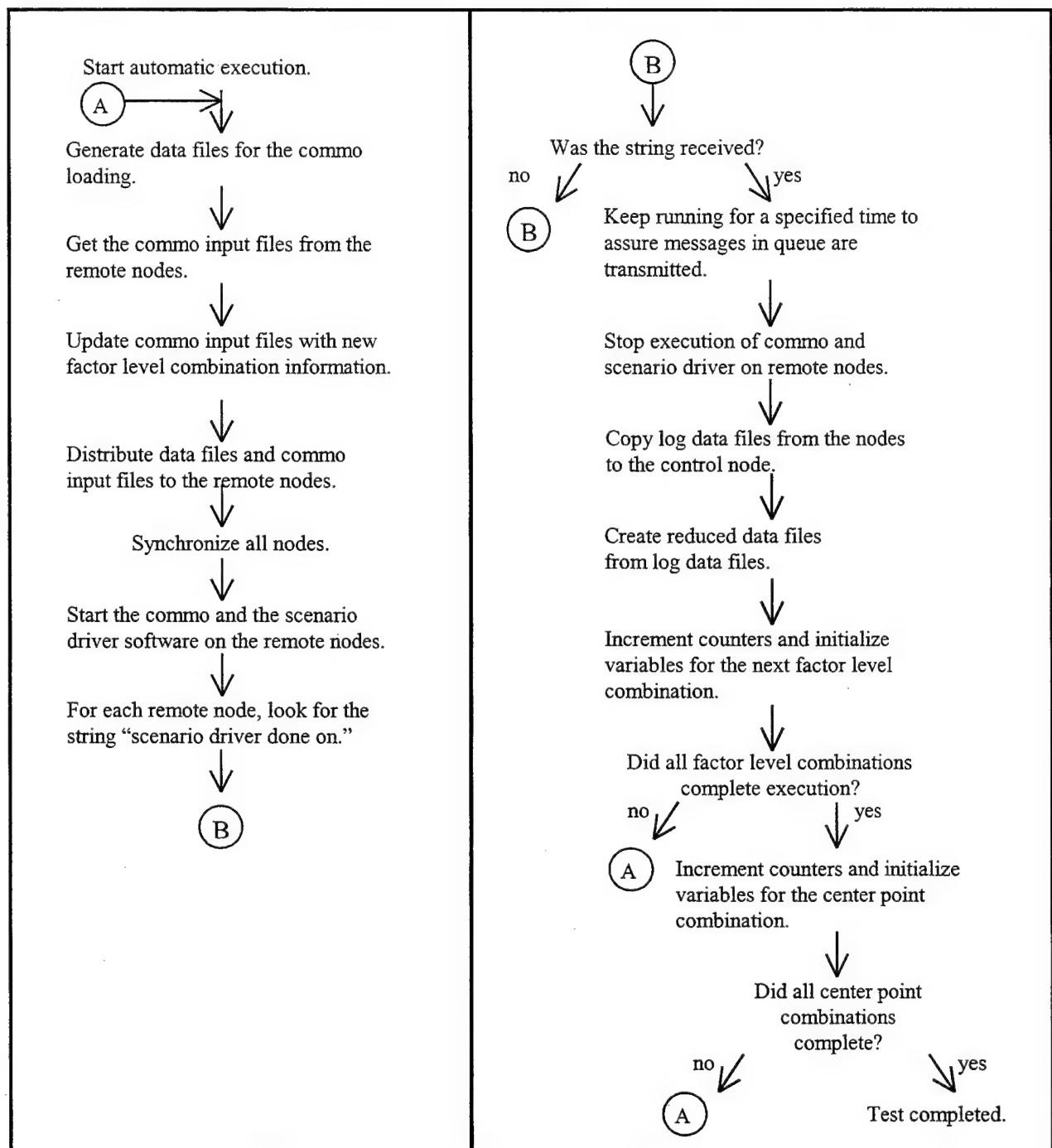


Figure 2. Flow of Events.

All the commands executed during a factor level combination are saved for later reference. A unique directory is created to save the data collected for each factor level combination, simplifying the search for a specific file.

For ARL's experiment, the following three main directories were used: (1) The local directory contained the **exp_driver** executable and its input files. (2) The DATA directory contained the template files. Directories for the output data files were created under the DATA directory. (3) The BIN directory contained the shell procedures.

Once the software executed, and as the factor level combinations were completed, unique directories were created for each replicate and factor level combination. Output data files and log files were copied into directories in which names reflected the replication and factor level combination for that run (i.e., for ARL's experiment, for replication 0, factor level combination 1, the information was automatically stored in the directory `./DATA/REP0/ITERATION1`).

4. Hardware Configuration

Different hardware configurations may be used to run an experiment with **exp_driver** as long as the input files contain the right information. The hardware configuration for the ARL experiment is explained in the next paragraph. The commo protocol and scenario driver were developed by ARL's researchers and programmers.

There were three remote nodes, each of which was a SPARCbook 3 [7]. Each contained a commo protocol and a scenario driver. The commo protocol included data collection functions to log the sending and receipt of messages and acknowledgments, as well as information on queues. The scenario driver provided the commo loading. The remote nodes were connected via ethernet to a SPARCstation 20 [8] that served as the data storage and control node (Figure 3). The remote nodes were connected to Single Channel Ground and Airborne Radio System (SINCGARS) combat net radios via tactical data buffers (TDBs) [9], a modem between the radios and the terminal equipment. Resistor loads were used as the antennas to reduce the transmission range. The TDB interfaced with the remote node using RS-232C, and with the SINCGARS using MIL-STD-188C(2) [10].

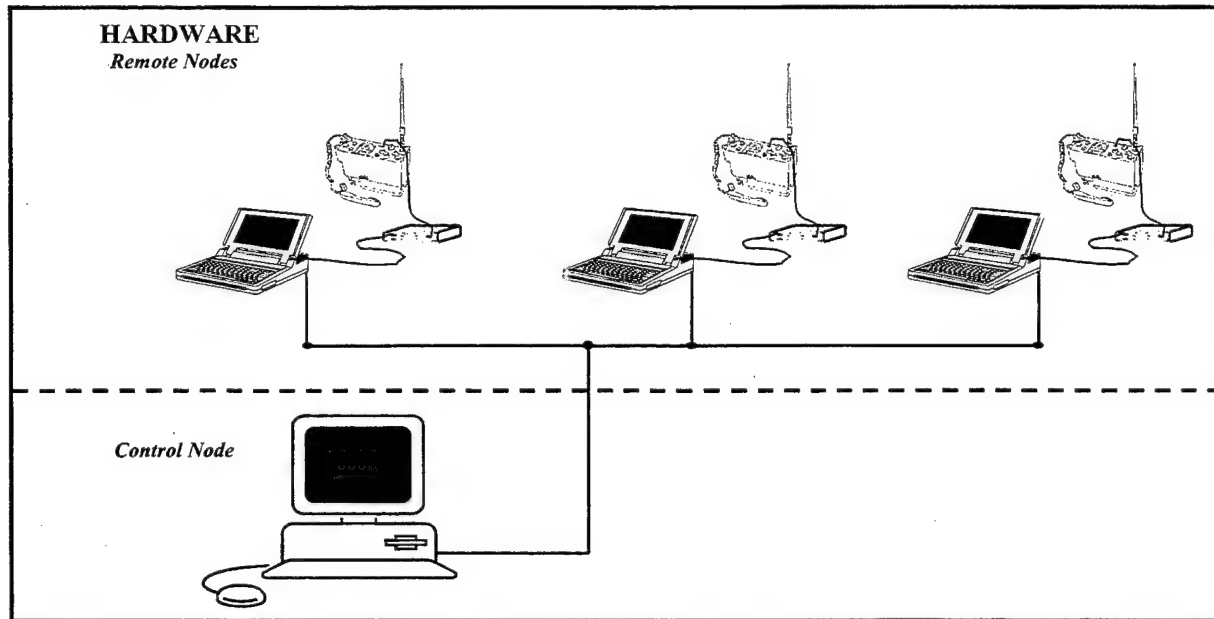


Figure 3. Example of a Hardware Configuration.

5. Conclusions

The template files are useful in simplifying the programmer's job when the experimental configuration requires modification. This allows the experimental configuration to be quickly and easily modified since the input is not "hard wired" in the code. For instance, if the number of nodes needs to be increased or decreased, the programmer modifies the input text files containing node information, and the updates on the remote software take place during the test driver initialization phase.

Because the test driver is of a general nature, it can be used in a variety of situations to run an experiment in a distributed UNIX environment. It is anticipated that future experiments can be automated to consider more complex communications protocol modifications. Automating the process reduces the chance of operator error and simplifies the execution of the experimental design.

INTENTIONALLY LEFT BLANK.

6. References

1. Kaste, V. A., A. E. Brodeen, and B. D. Broome. "An Experiment to Examine Protocol Performance Over Combat Net Radios." BRL-MR-3978, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, 1992.
2. Montgomery, D. C. *Design And Analysis of Experiments*. 3rd Edition, NY: John Wiley & Sons, 1991.
3. Kernighan, B. W., and D. M. Ritchie. *The C Programming Language*. 2nd Edition, Englewood Cliffs, NJ: Prentice-Hall, 1988.
4. Nye, A. *Xlib Reference Manual*. Volume 2, 3rd Edition, Sebastopol, CA: O'Reilly & Associates, 1992.
5. Ferguson, P. M. *Motif Reference Manual*. Volume 6B, 1st Edition, Sebastopol, CA: O'Reilly & Associates, 1994.
6. McGilton, H., and R. Morgan. *Introducing the UNIX System*. NY: McGraw-Hill, 1983.
7. Tadpole Technology, Inc. *SPARCbook 3 Series Technical Reference Manual*. Austin, TX, 1994.
8. Sun Microsystems Inc. *SPARCstation 20 System Specifications Manual*. Mountain View, CA, 1994.
9. Harris Corporation. *Harris RF Communications RF-3490 Digital Data Buffer Instruction Manual*. Rochester, NY, 1990.
10. U.S. Department of Defense. *Military Communication System Technical Standards*. MIL-STD-188C(2), Washington, DC, 1989.

INTENTIONALLY LEFT BLANK.

<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	OSD OUSD(A&T)/ODDDR&E(R) R J TREW THE PENTAGON WASHINGTON DC 20301-7100
1	DPTY CG FOR RDA US ARMY MATERIEL CMD AMCRDA 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN PO BOX 202797 AUSTIN TX 78720-2797
1	DARPA B KASPAR 3701 N FAIRFAX DR ARLINGTON VA 22203-1714
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE MADN MATH MAJ HUBER THAYER HALL WEST POINT NY 10996-1786
1	DIRECTOR US ARMY RESEARCH LAB AMSRL D D R SMITH 2800 POWDER MILL RD ADELPHI MD 20783-1197

<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>
1	DIRECTOR US ARMY RESEARCH LAB AMSRL DD 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB AMSRL CI AI R (RECORDS MGMT) 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB AMSRL CI AP 2800 POWDER MILL RD ADELPHI MD 20783-1197
	<u>ABERDEEN PROVING GROUND</u>
4	DIR USARL AMSRL CI LP (BLDG 305)

<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>
1	US ARMY RSRCH LAB AMSRL D J W LYONS ADELPHI MD 20783-1197
1	US ARMY RSRCH LAB WSMR TECH LIBRARY BR STEWIS IM ITZ WSMR NM 88002
25	US ARMY RSRCH LAB AMSRL CI CT M C LOPEZ APG MD 21005

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project(0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2001		3. REPORT TYPE AND DATES COVERED Final, June 1998 - June 1999
4. TITLE AND SUBTITLE Description of an Automated Tool for Evaluating a Tactical Communications Protocol			5. FUNDING NUMBERS 622783.Y109TED10	
6. AUTHOR(S) María C. López				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRL-CI-CT Aberdeen Proving Ground, MD 21005-5067			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-MR-509	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>Adaptive networking (AN) focuses on creating a logical and extendible framework for a multiyear effort in AN research. The U.S. Army Research Laboratory (ARL) designed an experiment to identify which communications protocol's parameters were more likely to have a relevant effect on network congestion. The software, called experiment control software (exp_driver), was developed to automatically execute the experimental design to test the protocol's parameters. Prior to the exp_driver existence, the experimental design for similar tests was executed with frequent operator intervention, requiring extra time for set up, and introducing human errors. This report provides a description of the exp_driver configuration for others who want to implement this software.</p>				
14. SUBJECT TERMS protocol "factorial design," software automated network tactical communication			15. NUMBER OF PAGES 16	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-MR-509 (López) Date of Report March 2001

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT
ADDRESS

Organization

Name

E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD
ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 0001, APG, MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
U.S. ARMY RESEARCH LABORATORY
ATTN AMSRL CI CT
ABERDEEN PROVING GROUND MD 21005-5067



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

